

# TP HAC

## Partie 1

**Consigne :** Utiliser un éditeur de texte pour écrire le code, exécuter votre script en console avec la commande (utiliser python 3) :

> python script.py

**Documentation :** UMTICE Introduction à la programmation - MPC/BBTE/Math-Eco, clé PYTHON2016

### Exercices :

**Exo 1 :** Ecrire un programme en Python qui demande à l'utilisateur de saisir un entier n compris entre 1 (inclus) et 100 (inclus). Le programme affiche ensuite le message approprié parmi les 3 possibilités suivantes :

- « Groupe A » si le nombre est pair et multiple de 3 ;
- « Groupe B » si le nombre est pair, mais pas multiple de 3 ;
- « Groupe C » si le nombre est nombre impair.

Utilisez l'opérateur modulo % (reste de la division entière) :

```
Entrez un entier compris entre 1 et 100 : 6
L'entier 6 appartient au groupe A
-----
Entrez un entier compris entre 1 et 100 : 10
L'entier 10 appartient au groupe B
```

Réécrire le programme, mais sans utiliser de « else »

**Exo 2 :** Ecrire le code Python (main et fonctions) permettant de saisir les coefficients d'une équation du second degré et d'afficher les solutions réelles quand elles existent.

**Exo 3 :** Ecrire de deux façons différentes le code qui vérifie s'il y a des entiers supérieurs à une valeur donnée dans une liste initialisée à [3,41,12,9,74,15] :

**Exo 4 :** Soient les listes suivantes :

```
[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
['Janvier', 'Février', 'Mars', 'Avril', 'Mai', etc.]
```

Écrivez un programme qui crée une nouvelle liste qui devra contenir tous les éléments des deux listes en les alternants, de telle manière que chaque nom de mois soit suivi du nombre de jours correspondant :

```
['Janvier', 31, 'Février', 28, 'Mars', 31, etc.]
```

**Exo 5 :** Même exercice que le 6, mais retourner un dictionnaire avec les mois en clé et le nombre de jours en valeur.

## Partie 2 : environnement de travail

**Vérifier si jupyter est déjà installé, si oui**

### Exo 1 : virtualenv

Nous allons créer un environnement virtuel de travail pour python. Commencer par lire le tutoriel suivant : <http://docs.python-guide.org/en/latest/dev/virtualenvs/>. Les commandes à utiliser sont :

```
mkdir p3env
cd p3env
virtualenv -p /usr/bin/python3 p3env
```

Activez votre environnement virtuel avec la commande :

```
source p3env/bin/activate
```

Note : Dans nos salles TP, *virtualenv* et *pip* pour *python3* sont déjà installés, mais il ne porte pas les mêmes noms (à vérifier).

### Exo 2 : notebook

Nous allons travailler dans un notebook dans la suite des TP. Un notebook permet d'exécuter du code python, d'afficher des résultats et de faire des commentaires au format markdown dans une page web.

1- Installer Jupyter

```
pip3 install jupyter
```

2 - Télécharger le notebook suivant (attention au extension) :

<https://nbviewer.jupyter.org/github/ipython/ipython/blob/master/examples/IPython%20Kernel/Custom%20Display%20Logic.ipynb#>

3 – executer le notebook

```
jupyter notebook CustumDisplayLogic.ipynb
```

4- Fait un restart du kernel en supprimant les affichages (clear all output).

5- Faite executer le notebook.

6- Cliquer sur l'icône jupyter, créer un notebook python 3 (menu new). Refaire l'exercice 4.

### Exo 3 : ajout d'outils

Installer les outils suivants :

```
pip3 install numpy
pip3 install matplotlib
pip3 install scipy
```

```
pip3 install sklearn
```

### Partie 3 : HAC (et K-mean)

Exo 1 : refaire en partie le tutoriel suivant :

<https://joernhees.de/blog/2015/08/26/scipy-hierarchical-clustering-and-dendrogram-tutorial/>

le but est d'avoir un dendrogramme avec troncature à partir du jeu de donnée proposé.

Exo 2 : refaire la même chose avec le jeu de données iris en utilisant l'exemple suivant :

<http://scikit-learn.org/stable/tutorial/basic/tutorial.html>

ou

[http://scikit-learn.org/stable/auto\\_examples/datasets/plot\\_iris\\_dataset.html](http://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html)

Iris : [https://en.wikipedia.org/wiki/Iris\\_flower\\_data\\_set](https://en.wikipedia.org/wiki/Iris_flower_data_set)

Exo 3 : tester différentes règles de mise à jour des distances (complet, single, ward). Afficher les dendrogrammes.

Exo 4 : programmer l'HAC